Kotlin - Strings

The Kotlin **String** data type is used to store a sequence of characters. String values must be surrounded by double quotes (" ") or triple quote (""" """).

We have two kinds of string available in Kotlin - one is called **Escaped String** and another is called **Raw String**.

- **Escaped string** is declared within double quote (" ") and may contain escape characters like '\n', '\t', '\b' etc.
- **Raw string** is declared within triple quote (""" """) and may contain multiple lines of text without any escape characters.

Example

```
fun main(args: Array<String>) {
   val escapedString : String  = "I am escaped String!\n"
   var rawString :String  = """This is going to be a
   multi-line string and will
   not have any escape sequence""";

   print(escapedString)
   println(rawString)
}
```

When you run the above Kotlin program, it will generate the following output:

```
I am escaped String!
This is going to be a
multi-line string and will
not have any escape sequence
```

This is optional to specify the data type for a String, Kotlin can understand that the a variable is a String because of the given double or tripple quotes.

If you want to create a String variable without assigning the value then you must specify the type while declaring the variable otherwise it will raise an error:

```
fun main(args: Array<String>) {
   val name : String

   name = "Zara Ali"

   println(name)
}
```

When you run the above Kotlin program, it will generate the following output:

```
Zara Ali
```

Kotlin String Templates

Kotlin string templates are pieces of code that are evaluated and whose results are interpolated into the string. A template expression starts with a dollar sign ($) and may consist of either a name or an expression.

```
fun main(args: Array<String>) {
   val name : String = "Zara Ali"

   println("Name  - $name")  // Using template with variable name

   println("Name length - ${name.length}")  // Using template with expression.
}
```

When you run the above Kotlin program, it will generate the following output:

```
Name - Zara Ali
Name length - 8
```

Kotlin String Object

Kotlin String is an object, which contains a number of properties and functions that can perform certain operations on strings, by writing a dot character (.) after the specific string variable.

We will see some of the important properties and functions in this chapter, remaining you can find in official documentation of Kotlin latest version.

Kotlin String Indexes

Kotlin String can be treated as a sequence of characters or you can say String is an array of characters. You can access its element by specifying the index of the element using a square brackets.

String indexes start with 0, so if you want to access 4th element of the string then you should specify index as 3 to access the 4th element.

Example

```
fun main(args: Array<String>) {
   val name : String = "Zara Ali"
```

```
    println(name[3])
    println(name[5])
}
```

When you run the above Kotlin program, it will generate the following output:

```
a
A
```

Kotlin String Length

We can use **length** property of Kotlin string to find out its length.

Kotlin function **count()** also returns the length of a given string.

Example

```
fun main(args: Array<String>) {
    val name : String = "Zara Ali"

    println("The length of name :" + name.length)
    println("The length of name :" + name.count())

}
```

When you run the above Kotlin program, it will generate the following output:

```
The length of name :8
The length of name :8
```

Kotlin String Last Index

We can use **lastIndex** property of Kotlin string to find out the index of the last character in the char sequence. If a string is empty then it returns a -1.

Example

```
fun main(args: Array<String>) {
    val name : String = "Zara Ali"

    println("The index of last character in name :" + name.lastIndex)
}
```

When you run the above Kotlin program, it will generate the following output:

```
The index of last character in name :7
```

Changing Case of Strings

Kotlin provides **toUpperCase()** and **toLowerCase()** functions to convert a string into upper case and lower case respectively.

Example

```
fun main(args: Array<String>) {
   val name : String = "Zara Ali"

   println("Upper case of name :" + name.toUpperCase())
   println("Lower case of name :" + name.toLowerCase())
}
```

When you run the above Kotlin program, it will generate the following output:

```
Upper case of name :ZARA ALI
Lower case of name :zara ali
```

Kotlin String Concatenation

We can use either **+** operator to concatenate two strings, or we can also use **plus()** function to concatenate two strings.

Example

```
fun main(args: Array<String>) {
   var firstName : String = "Zara "
   var lastName : String = "Ali"

   println("Full Name :" + firstName + lastName)

   println("Full Name :" + firstName.plus(lastName) )
}
```

When you run the above Kotlin program, it will generate the following output:

```
Full Name :Zara Ali
Full Name :Zara Ali
```

Trim Characters from a String

We can remove first few or last few characters from a string using **drop()** or **dropLast()** functions.

Example

```
fun main(args: Array<String>) {
   var name : String = "Zara Ali"

   println("Remove first two characters from name : " + name.drop(2))
```

```
    println("Remove last two characters from name : " + name.dropLast(2))
}
```

When you run the above Kotlin program, it will generate the following output:

```
Remove first two characters from name : ra Ali
Remove last two characters from name : Zara A
```

Quotes Inside a String

To use quotes inside a string, use single quotes ('):

Example

```
fun main(args: Array<String>) {
   var str1 : String = "That's it"
   var str2 : String = "It's OK"

   println("str1 : " + str1)
   println("str2 : " + str2)
}
```

When you run the above Kotlin program, it will generate the following output:

```
str1 : That's it
str2 : It's OK
```

Finding a String inside a String

Kotlin provides **indexOf()** function to find out a text inside a string. This function returns the index of the first occurrence of a specified text in a string

Example

```
fun main(args: Array<String>) {
   var str : String = "Meditation and Yoga are synonymous with India"

   println("Index of Yoga in the string - " + str.indexOf("Yoga"))
}
```

When you run the above Kotlin program, it will generate the following output:

```
Index of Yoga in the string - 15
```

Comparing Two Strings

Kotlin provides **compareTo()** function to compare two strings. This function returns 0 if two strings are equal otherwise it will return 1.

Example

```
fun main(args: Array<String>) {
   var str1 : String = "Apple"
   var str2 : String = "Apple"

   println(str1.compareTo(str2))
}
```

When you run the above Kotlin program, it will generate the following output:

```
0
```

Kotlin getOrNull() function

Kotlin **getOrNull()** function returns a character at the given index or null if the index is out of bounds of this char sequence.

Example

```
fun main(args: Array<String>) {
   var name : String = "Zara"

   println(name.getOrNull(0))
   println(name.getOrNull(2))
   println(name.getOrNull(100))
}
```

When you run the above Kotlin program, it will generate the following output:

```
Z
r
null
```

Kotlin toString() function

Kotlin **toString()** function returns a string representation of the object..

Example

```
fun main(args: Array<String>) {
   var name : String = "Zara Ali"

   println(name.toString())
}
```

When you run the above Kotlin program, it will generate the following output:

```
Zara Ali
```

Quiz Time (Interview & Exams Preparation)

Q 1 - Which of the following is true about Control Flow Statement?

A - Control flow controls the execution of the program

B - Loops and Decision Statements are part of control flow

C - Control flow is an essential part of modern programming languages

D - All of the above

Q 2 - Which of the following is a control flow statement in Kotlin?

A - String

B - Fun

C - When

D - None of the above

Q 3 - If we do not have control flow statements, then it will be almost impossible to write a computer program?

A - True

B - False